Electiva

**Marco Teórico**

**Capítulo 1. Historia de la empresa**

Esta empresa fue abierta en diciembre del 1995 en la casa del señor **Saluterio Portorreal** haciendo solo arreglos a ropa, luego con experiencia empezó a fabricar camisas, pantalones, chaquetas, etc.

La empresa crecio y se convirtió en una gran sastrería con varios empleados, entre ellos uno llamado Jesus Minier, el cual el día de hoy es un político del área, también estuvo uno llamado Richard Nin, el cual aprendió y puso su propia sastrería, solo hace reparaciones.

Al pasar el tiempo el propietario compro los accesorios para implementar una lavandería, la cual creció y hoy en día está en buen funcionamiento.

**La necesidad de automatizar el proceso**

En la empresa ha habido muchos inconvenientes con la entrega de productos porque no se les da a los clientes el lugar que corresponden, me explico, si alguien va a lavar y/o planchar y viene otro a hacer lo mismo, entonces se toma el que más urgencia tenga, sin saber si el otro vendrá de una vez o tardara un poco más.

Con este sistema se llevará el control de entrada y salida de productos en el mismo orden que estén listo los productos, siempre y cuando tenga su tiempo preestablecido.

**Tema II – Que hace el Sistema**

**Que se hacía antes de implementar el sistema**

Al momento de recibir una prenda el encargado le crea un recibo a mano, en el cual dice la cantidad de prendas recibidas, que se le va hacer (lavar o planchar), el costo de cada una y el total.

Estas prendas son pasadas al encargado de plancha o lavado y luego de ser planchadas o lavadas son puestas en el estante de listas para entrega.

Cuando el cliente venía a buscar su prende con el número de recibo se buscan y luego se entregan.

**Ahora con el sistema**

Lo que buscamos es hacer que nuestros clientes se sientan más confiados con este sistema y sepan en el tiempo estipulado con el mismo serán entregadas sus prendas.

**Sistema de sastrería para P&L v.4.15**

**Justificación**

Crear una solución informática para automatizar el proceso de:

* Recibir.
* Lavar y/o planchar
* Entregar

Al principio no se necesitaba un sistema automatizado, ya que en el negocio solo se hacían reparaciones de ropa, y uno que otro pantalón, pero al ir creciendo y pasar de una simple sastrería a una lavandería, las cosas cambiaron.

Este programa viene a solucionar lo que es el proceso lavado y planchado fuera de orden y que los clientes puedan ver el estatus de sus productos en línea. El programa crea una numeración por clientes y una contraseña para llegar a ver sus dichos estatus.

**1 pagina**

**“Hablar acerca de mi producto”**

**Objetivos generales y específicos**

**Generales**

Crear una aplicación que ayude a mejorar el desarrollo de este sistema de entrada, hacer y devolver en menos tiempo y con más calidad y claridad.

**Específicos**

* Llevar un control de las entradas y salidas de productos de este local.
* Poder ver el estatus de su producto en línea.
* Reducir el tiempo de entrega.
* Llevar control del flujo de efectivo.
* Controlar el gasto de materia prima en el negocio.

**Plataforma**

WAMP

Este tipo de plataforma web suele usarse como herramienta de desarrollo por la mayoría de los programadores que tienen Windows como SO principal y que desarrollan aplicaciones web en PHP. Hacemos pruebas de nuestra aplicación PHP en el servidor de desarrollo WAMP y luego la subimos al servidor LAMP. Existe un software llamado WAMP5 que instala de forma compacta y configura fácilmente toda la plataforma WAMP.

Ventajas: Ideal para utilizar como servidor de desarrollo. ′ Fácil instalación con el software WAMP5. Desventajas: Puedes tener alguna dificultad a la hora de instalar los subsistemas por separado. ′ Puede que siempre se te dificulte la instalación de PHP en IIS (Internet Information Server).

Esta plataforma web promocionado por el editor de la editorial O’Reilly dale Dougherty y algunos auntores de MySQL es el más popular y el que podrás encontrar mas facilmente en la mayoría de los proveedores de Hosting.

### Lenguaje PHP

Es un lenguaje de programación utilizado para la creación de sitio web. [PHP](http://www.maestrosdelweb.com/editorial/phpintro/) es un acrónimo recursivo que significa “PHP Hypertext Pre-processor”, (inicialmente se llamó Personal Home Page). Surgió en 1995, desarrollado por PHP Group.

PHP es un lenguaje de script interpretado en el lado del servidor utilizado para la generación de páginas web dinámicas, embebidas en páginas HTML y ejecutadas en el servidor. PHP no necesita ser compilado para ejecutarse. Para su funcionamiento necesita tener instalado Apache o IIS con las librerías de PHP. La mayor parte de su sintaxis ha sido tomada de C, Java y Perl con algunas características específicas. Los archivos cuentan con la extensión (php).

Este lenguaje es ayudado por CSS.

[CSS](http://es.wikipedia.org/wiki/Css) es un lenguaje de estilo que define la presentación de los documentos [HTML](http://es.wikipedia.org/wiki/Html). Por ejemplo, CSS abarca cuestiones relativas a fuentes, colores, márgenes, líneas, altura, anchura, imágenes de fondo, posicionamiento avanzado y muchos otros temas. ¡Espera unos segundos y ya verás!

Es posible usar HTML, o incluso abusar del mismo, para añadir formato a los sitios web. Sin embargo, CSS ofrece más opciones y es más preciso y sofisticado. CSS está soportado por todos los navegadores hoy día.

**Bootstrap**

**Bootstrap** es una biblioteca multiplataforma o conjunto de herramientas de [código abierto](https://es.wikipedia.org/wiki/C%C3%B3digo_abierto) para diseño de sitios y aplicaciones web. Contiene plantillas de diseño con tipografía, formularios, botones, cuadros, menús de navegación y otros elementos de diseño basado en [HTML](https://es.wikipedia.org/wiki/HTML) y [CSS](https://es.wikipedia.org/wiki/Hojas_de_estilo_en_cascada), así como extensiones de [JavaScript](https://es.wikipedia.org/wiki/JavaScript) adicionales. A diferencia de muchos frameworks web, solo se ocupa del desarrollo [front-end](https://en.wikipedia.org/wiki/Front-end_web_development).

Procesador y bus de datos atron, celeron Intel 32 o 64 bit

Framework

Todo lo relacionado con el ambiente es plataforma

* Microsoft
* Google
* Apple
* Motorolla

**Ambiente**

Web

ayudado por

HTML5

Php

Mysql

Bluehost

Domain.com

Esta aplicación es creada para correr en WAMP, será usada para dar orden a la llegada de los clientes, para que el cliente pueda ver el estatus de su producto en línea, se ejecutara a nivel web (no tendrá otra aplicación donde como esclavo o APK para ser instalada).

Esta página será dinámica, ya que mostrar la información que los clientes están buscando (El estatus de su mercancía).

Decir a que pertenece eje, ambientes web de aplicaciones, decir versión y subversión.

## Versiones por número.

Algo común es realizar el manejo de versiones mediante 3 números: X.Y.Z y cada uno indica una cosa diferente:

* El primero (X) se le conoce como **versión mayor** y nos indica la versión principal del software. Ejemplo: 1.0.0, 3.0.0
* El segundo (Y) se le conoce como **versión menor** y nos indica nuevas funcionalidades. Ejemplo: 1.2.0, 3.3.0
* El tercero (Z) se le conoce como **revisión** y nos indica que se hizo una revisión del código por algun fallo. Ejemplo: 1.2.2, 3.3.4

Esta versión como es la primera será la 1.0.0

**Arquitectura**

# Arquitectura Orientada a Servicios (SOA)

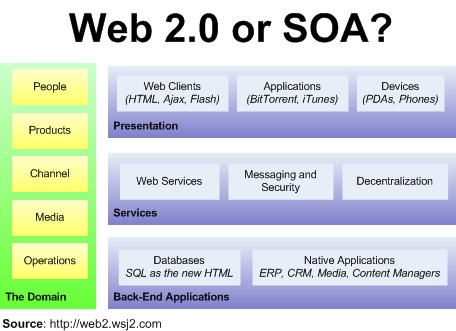
 Ejemplo: implementación de SOA en la empresa El proceso de servicio de pedidos en Northern Electronics comienza con la apertura de una orden de pedido, y sigue con una consulta al inventario para comprobar si los artículos existen en stock. Finalmente, dentro de un proceso colaborativo, se organiza una expedición de transporte para la carga y entrega del material al cliente. Cada una de estas partes del proceso se compone de varios pasos no automatizados aún.

 [4.](https://image.slidesharecdn.com/ejemplosoa-160720231225/95/ejemplo-soa-4-638.jpg?cb=1469056396) Northern Electronics decide pasar de las órdenes de pedido escritas en papel a los formularios electrónicos, automatizando el proceso de transferencia de información entre los departamentos de ventas y logística. Logística verifica el stock: si no hay unidades de los artículos solicitados, la orden de pedido (PO) se pone como “pendiente” y los artículos se marcan también. Si los artículos existen en el almacén, un empleado del departamento de logística contacta con la empresa de transporte y luego entrega al encargado de la manipulación en el almacén la lista de los artículos y cantidades que deben sacarse de sus estanterías para la preparación de los embalajes. Northern Electronics decide que estas dos partes de la fase de cumplimentación de pedidos son dos buenas candidatas para su automatización y conversión en servicios Web. Exposición Northern Electronics ha decidido implementar una capa orientada a servicios flexible y reutilizable que exponga la lógica de negocio de las aplicaciones actuales (y de sus partners), a fin de reducir las ineficiencias detectadas en los procesos manuales de servicio de pedidos y notificación de estado. Estos son algunos ejemplos: • ShippingService – que es el servicio web que utilizan en Northern Electronics para enviar y recibir detalles de la manipulación del pedido. • PickupService – que es el proceso interno de Northern Electronics para notificar al responsable de la carga en el almacén la lista de productos que componen la expedición y para confirmar que esta manipulación se ha realizado. • TransportService – que es el servicio web de la empresa de transportes, que se utiliza para gestionar las expediciones y confirmar las salidas y entregas de material. Esta parte de la solución basada en servicios Web debe implementarla la empresa de transportes. Cada uno de estos servicios de negocio lleva debajo distintos servicios de grano fino o llamadas al API que se combinan para generar una capa de servicios de mayor alcance. Esta capa de

 [5.](https://image.slidesharecdn.com/ejemplosoa-160720231225/95/ejemplo-soa-5-638.jpg?cb=1469056396) servicios se implementa en Northern Electronics utilizando desarrollos propios basados en .NET Framework y Visual Studio y también adaptadores de integración externos para sus aplicaciones de línea de negocio (LOB) y sistemas de mainframe.. Composición Cuando ya se puede acceder a las aplicaciones en forma de servicios, Northern Electronics hace la correspondencia entre las distintas fases de sus procesos de negocio y el modelo de procesos orquestado que organiza, controla y monitoriza todas y cada una de las etapas del flujo de trabajo de la gestión de pedidos. Un principio estratégico fundamental para ellos es no escribir sus reglas y lógica de negocio dentro del código de implementación, sino externalizar dicha lógica en un modelo que pueda, en cualquier momento, verse y modificarse con facilidad. Con esta medida se gana un mayor grado de visibilidad y trazabilidad, e incluso la posibilidad de añadir métricas de procesos en tiempo real y eventos dentro de un panel de decisiones o cuadro de mandos. Se puede monitorizar el estado de salud del proceso de principio a fin, y no solamente cuando las aplicaciones individuales han finalizado su labor, y además se puede hacer un seguimiento sobre el conjunto completo de actividades que componen el proceso de negocio. Si algún paso dentro del proceso falla, por ejemplo un retraso en la llegada de una expedición, los usuarios pueden recibir notificaciones de inmediato, y adoptar las acciones correctoras necesarias. Esta capa de composición flexible está implementada en Northern Electronics utilizando BizTalk Server y Visual Studio para modelizar el proceso de negocio y definir la lógica de orquestación; con SQL Server y BizTalk Server se hace el seguimiento de los procesos en tiempo real, utilizando BAM (Business Activity Monitoring) y con Microsoft Office SharePoint Server se automatizan los procesos de colaboración de tipo manual y se asignan tareas del workflow a los usuarios durante el proceso. Consumo Puesto que los usuarios finales no son expertos en sistemas de

 [6.](https://image.slidesharecdn.com/ejemplosoa-160720231225/95/ejemplo-soa-6-638.jpg?cb=1469056396) información, Northern Electronics ha decidido crear una aplicación compuesta basada en Office (se la suele denominar una “Aplicación de Negocio de Office”, OBA por sus iniciales en ingles). Esta OBA permite a los usuarios accede fácilmente a la información y procesos que necesitan, utilizando unas herramientas que conocen bien y a las que están acostumbrados. Este hecho mejora la productividad y evita la necesidad de emprender un programa de formación de gran envergadura para los usuarios finales. Los formularios utilizados en los departamentos de ventas y logística están creados con InfoPath, una solución para la captura dinámica de información. Estos formularios de InfoPath se guardan y se gestionan dentro de Microsoft Office SharePoint Server, y sus capacidades internas de gestión del workflow documental les permiten automatizar de forma sencilla el enrutamiento de los formularios entre los distintos departamentos y equipos de trabajo. Conforme se crean nuevos flujos de trabajo o se asignan tareas, estas actividades quedan asignadas automáticamente a los usuarios mediante correos de Outlook y su seguimiento se efectúa mediante las colas de tareas de SharePoint. Si se da la circunstancia de que los workflows de colaboración alojados dentro de SharePoint tengan que acceder a sistemas de backend estructurados, la integración nativa de SharePoint y BizTalk Server permite invocar la capa de servicios de negocio o la lógica de orquestación de los procesos. Los resultados El primer proyecto de implementación se acotó de forma muy explícita y tras su puesta en marcha se obtuvieron retornos de inversión a los 90 días. Aun después de completarse esta “tajada” inicial de la infraestructura SOA, Northern Electronics ha seguido capitalizando estos beneficios. Sin tener que esperar a un gran proyecto de implementación de infraestructuras de varios años de duración, ya pueden, casi al momento, gestionar su cadena de valor de forma más eficiente y son capaces de identificar y aprovechar nuevas oportunidades para consolidar y automatizar actividades internas dentro de su cadena de valor.

hacer una grafica

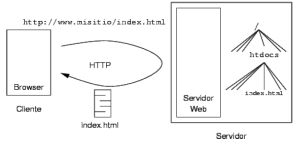


y explicar el funcionamiento

Una aplicación Web  es  proporcionada  por  un  servidor  Web  y  utilizada  por  usuarios  que  se Conectan desde cualquier punto vía clientes Web (browsers o navegadores). La arquitectura de un Sitio Web tiene tres componentes principales:

* Un servidor Web
* Una conexión de red
* Uno o más clientes

El servidor Web distribuye páginas de información formateada a los clientes que las solicitan. Los requerimientos son hechos a través de una conexión de red, y para ello se usa el protocolo HTTP. Una vez que se solicita esta petición mediante el protocolo HTTP y la recibe el servidor Web, éste localiza la página Web en su sistema de archivos y la envía de vuelta al navegador que la solicitó.

[](https://programacionwebisc.files.wordpress.com/2013/11/unidad2.png)

Las aplicaciones Web están basadas en el modelo Cliente/Servidor que gestionan servidores web, y que utilizan como interfaz páginas web.

Las páginas Web son el componente principal de una aplicación o sitio Web. Los browsers piden páginas  (almacenadas  o creadas  dinámicamente) con  información  a  los  servidores  Web.  En algunos  ambientes  de  desarrollo de  aplicaciones  Web, las  páginas  contienen  código  HTML  y scripts dinámicos, que son ejecutados por el servidor antes de entregar la página.

Una vez que se entrega una página, la conexión entre el browser y el servidor Web se rompe, es decir que la lógica del negocio en el servidor solamente se activa por la ejecución de los scripts de las páginas solicitadas por el browser (en el servidor, no en el cliente). Cuando el browser ejecuta un script en el cliente, éste no tiene acceso directo a los recursos del servidor. Hay  otros  componentes  que no  son  scripts, como los  applets  (una aplicación  especial  que  se ejecuta  dentro de  un  navegador)  o los  componentes  ActiveX. Los  scripts  del  cliente  son  por lo general código JavaScript o VBSscript, mezclados con código HTML.

La colección de páginas son en una buena parte dinámicas (ASP, PHP, etc.), y están agrupadas lógicamente para dar un servicio al usuario. El acceso a las páginas está agrupado también en el tiempo (sesión). Los componentes de una aplicación Web son:

1.  Lógica de negocio.

* Parte más importante de la aplicación.
* Define los procesos que involucran a la aplicación.
* Conjunto de operaciones requeridas para proveer el servicio.

2.  Administración de los datos.

* Manipulación de BD y archivos.

3.  Interfaz

* Los usuarios acceden a través de navegadores, móviles, PDAs, etc.
* Funcionalidad accesible a través del navegador.
* Limitada y dirigida por la aplicación.

Las aplicaciones web se modelan mediante lo que se conoce como modelo de capas,  Una capa representa un elemento que procesa o trata información. Los tipos son:

* Modelo  de  dos  capas: La  información  atraviesa  dos  capas  entre  la interfaz  y  la administración de los datos.
* Modelo de n-capas: La información atraviesa varias capas, el más habitual es el modelo de tres capas.

**Modelo de dos Capas.**

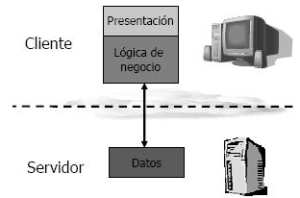
Gran parte de la aplicación corre en el lado del cliente (fat client).

Las capas son:

* Cliente (fat client): La lógica de negocio está inmersa dentro de la aplicación que realiza el interfaz de usuario, en el lado del cliente.
* Servidor: Administra los datos.

Las limitaciones de este modelo son.

* Es difícilmente escalable
* Número de conexiones reducida
* Alta carga de la red.
* La flexibilidad es restringida
* La funcionalidad es limitada.

[](https://programacionwebisc.files.wordpress.com/2013/11/mod2capas.png)

**Modelo de tres Capas.**

Esta diseñada para superar las limitaciones de las arquitecturas ajustadas al modelo de dos capas, introduce una capa intermedia (la capa de proceso) Entre  presentación y  los  datos, los  procesos  pueden  ser manejados  de forma separada a  la  interfaz de usuari o y a los datos, esta capa intermedia centraliza la lógica de negocio, haciendo la administración  más  sencil a, los  datos  se pueden integrar de  múltiples  fuentes, las  aplicaciones web actuales se ajustan a este modelo.

Las capas de este modelo son:

1.  Capa de presentación (parte en el cliente y parte en el servidor)

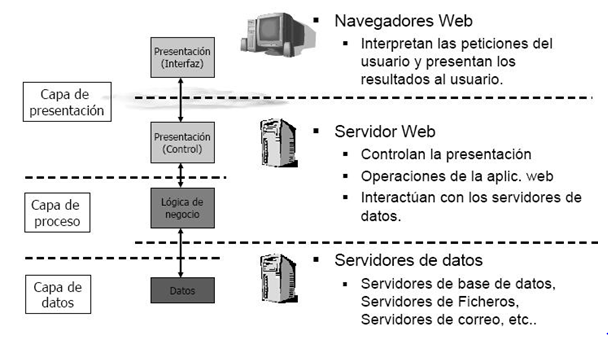
* Recoge la información del usuario y la envía al servidor (cliente)
* Manda información a la capa de proceso para su procesado
* Recibe los resultados de la capa de proceso
* Generan la presentación
* Visualizan la presentación al usuario (cliente)

2.  Capa de proceso (servidor web)

* Recibe la entrada de datos de la capa de presentación
* Interactúa con la capa de datos para realizar operaciones
* Manda los resultados procesados a la capa de presentación

3.  Capa de datos (servidor de datos)

* Almacena los datos
* Recupera datos
* Mantiene los datos
* segura la integridad de los datos

[](https://programacionwebisc.files.wordpress.com/2013/11/3cap.png)

Anuncios

Informar de este anuncio

Anuncios

**Modelo y metodología**

**Método agile: Scrum**

El 11 de febrero del 2001, diecisiete gurús del mundo del desarrollo de software ya cansados de la forma tradicional de desarrollo en donde predomina la documentación, los controles de cambio burocráticos y las jerarquías, se reunieron para buscar nuevas alternativas con las cuales se pudieran generar productos de más valor, en tiempo y sobre todo, que el proceso de creación fuera motivante e interesante para todos.

El resultado de este viaje fue un manifiesto al que llamarón: “**el Manifiesto Ágil**”. Este manifiesto que consta de 4 valores, fue el compromiso que hicieron estos diecisiete gurús y lo hicieron escribiendo lo siguiente:

**Individuos e interacciones sobre procesos y herramientas**

Este primer valor nos habla de **la importancia que tienen las personas y sus interacciones**. Un ejemplo muy común, contrario a este valor, es mandar correos a la persona que tenemos a nuestro lado o mandarle mensajes por el chat de la empresa en lugar de la comunicación cara a cara, directa y efectiva, y que aunque suene impresionante, esto ocurre realmente todos los días en las organizaciones.

**Software que funciona sobre documentación exhaustiva**

La métrica / indicador más valioso en el mundo de la agilidad es el “software que funciona”. Es de gran importancia contar con algo tangible que pueda ayudar a los usuarios a tomar decisiones importantes en cuanto a su producto. Este indicador también es clave para los equipos desarrollo ya que les da claridad y elimina ambigüedades gracias al feedback temprano que los usuarios.

**Colaboración con el cliente sobre negociación de contratos**

Punto controversial en donde las áreas de sistemas se cuidan porque “el cliente no sabe lo que quiere” y las áreas del negocio se cuidan porque “sistemas es lento y hace las cosas sin calidad”. Necesitamos comenzar a derribar los muros entre el negocio y sistemas y comenzar a colaborar como un mismo equipo en donde la persona de la idea es parte del equipo de desarrollo y trabajan juntos para poder lograr generar valor al negocio.

**Responder al cambio sobre seguimiento a un plan**

Si queremos sobrevivir en este mundo **VUCA** (Volátil, Incierto, Complejo y Ambiguo) en el que vivimos es de suma importancia adaptarnos a los cambios. La tecnología cambia todos los días, el mercado, los competidores, las necesidades, prácticamente todo está en constante cambio y es por esto que debemos estar abiertos a cambiar y a tratar de dejar de controlarlo todo.

**Product Ouner**

El dueño de la empresa

**Scrum master**

Donaldo Artiles

**Team Members**

Donaldo Artiles

Ramón Plata

Neifi García

**Agile leaders**

Ramon Plata

**Patrón de diseño de desarrollo**

**Tiene que estar a cordee a la arquitectura.**

**Prototipo o mockup**

**Repositorio con git**